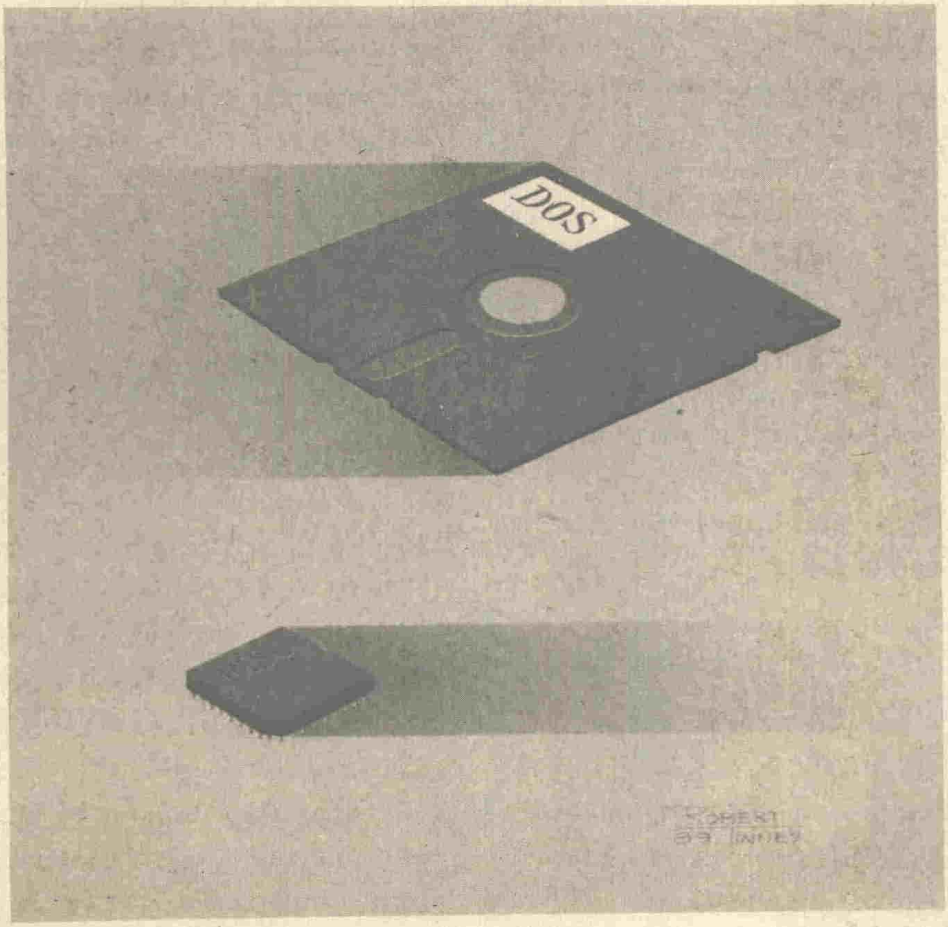
RISC & SPARC
พงศ์พิชญ์ พะหุชนม์

จาก RISC ถึง DOS

กรรมวิธี 2 แบบ ที่สามารถจะโอนย้าย โปรแกรมจาก RISC ไปใช้งานบน DOS



ไมโครโปรเซสเซอร์ 32 บิต เช่น 80386, 68030, และ RISC ได้ก่อให้เกิดการเปลี่ยนแปลงโครงสร้างครั้งใหญ่ของเครื่องคอมพิวเตอร์ ซึ่งทำให้เครื่องคอมพิวเตอร์มีขนาดประสิทธิภาพสูงขึ้นอย่างมาก ตั้งแต่เวอร์กสเตชัน เครื่องระดับกลาง และเครื่องขนาดใหญ่ที่มีผู้ใช้หลายคน แต่การเปลี่ยนแปลงนี้กลับมีผลน้อยมากต่อเครื่องชนิดตั้งโต๊ะที่ใช้ 8088 และ 80286 ซึ่งเป็นเครื่องที่มีมากที่สุด

แม้ว่าเครื่องที่ใช้ RISC จะเป็นเครื่องที่มีประสิทธิภาพสูงสุดในบรรดาเครื่องชนิดตั้งโต๊ะด้วยกัน แต่กลับได้รับการยอมรับน้อยมากเนื่องจากขาดแอพพลิเคชันในการใช้งานด้านต่าง ๆ ผู้ใช้จำนวนนับล้าน ๆ คนยังเคยชินกับการใช้โปรแกรมที่ทำงานบนดอสอยู่ เช่น สเปรดชีต เวิร์ดโปรเซสเซอร์ ระบบฐานข้อมูล และโปรแกรมอื่น ๆ ซึ่งผู้ใช้เหล่านี้จะไม่หันมาใช้ RISC ถ้าพวกเขาไม่สามารถนำโปรแกรมที่ใช้อยู่บนดอสมาใช้บน RISC ได้ เพราะความคุ้นเคยกับดอสมากเกินกว่าที่จะเรียนรู้โปรแกรมใหม่ หนทางที่จะแก้ปัญหานี้ก็คือ ต้องทำให้โปรแกรมบนระบบดอสสามารถทำงานบนเครื่องที่ใช้ RISC ได้

มันไม่ง่ายเลยที่จะทำให้โปรแกรมที่ทำงานบนดอสสามารถทำงานบนเครื่องที่ใช้ RISC ได้ และวิธีการที่ใช้อยู่ในปัจจุบันก็มีข้อเสียหลายอย่าง

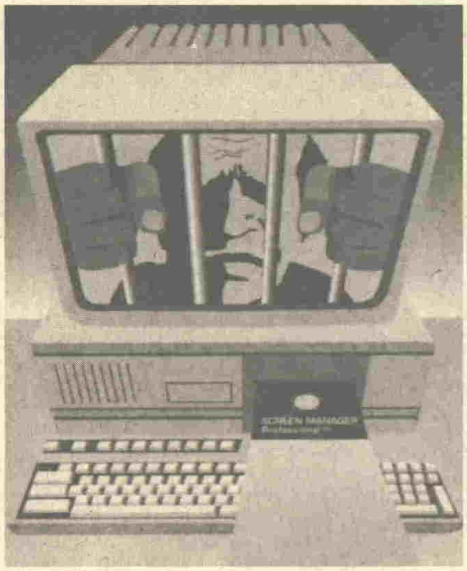
แต่มีวิธีใหม่ที่เรียกว่า ไบนารีพอร์ตติ้ง (binary porting) ซึ่งจะทำการเปลี่ยนจากภาษาเครื่องของ 8086 ของโปรแกรมที่ทำงานบนดอสเป็นภาษาเครื่องของ RISC โดยตรง ซึ่งจะทำให้สามารถนำโปรแกรมที่ทำงานบนดอสมาใช้บนเครื่องที่ใช้ RISC ได้

เรื่องของ RISC

ซีพียู RISC (Reduce Instruction Set Computer) ตัวแรกได้ถูกพัฒนาขึ้นในช่วงปลายทศวรรษ 1970 ถึงต้นทศวรรษ 1980 โดย IBM, มหาวิทยาลัย USC Berkeley และมหาวิทยาลัยสแตนฟอร์ด โดยมีแต่คำสั่งที่จำเป็นต่อการใช้งานพื้นฐานเท่านั้น มันจึงมีชุดคำสั่งน้อย ทำให้สามารถลดความซับซ้อนของโครงสร้างและทำงานได้เร็วขึ้น

แนวความคิดของ RISC ได้ถูกนำมาใช้ในซีพียูที่ผลิตในเชิงพาณิชย์หลายตัวเช่น 88000 ของโมโตโรล่า, R2000 และ R3000 ของ MIPS, Sun SPARC ซึ่งซีพียูเหล่านี้จะทำงานในแต่ละคำสั่งโดยใช้จำนวน machine cycle น้อยที่สุดเท่าที่เป็นไปได้ และส่วนมากจะสามารถทำงานได้ภายในหนึ่งไซเกิล

RISC คอมพิวเตอร์จะใช้ออปติไมซิงคอมพายเลอร์ (optimizing compiler) เพื่อแปลเป็นภาษาเครื่องที่กะทัดรัดทำงานได้เร็ว และยังลดการเสียเวลาในไปป์ไลน์ ซึ่งจะทำให้เครื่องคอมพิวเตอร์ที่ใช้ RISC สามารถทำงานได้เร็วกว่าเครื่องคอมพิวเตอร์แบบเดิมมาก ถึงแม้ว่าโปรแกรมของ RISC บางครั้งอาจจะยากกว่าโปรแกรมเดิม



แม้ว่า RISC จะทำงานเร็วกว่ามาก แต่มันก็ประสบปัญหาขาดแคลนโปรแกรมใช้งาน และการย้ายโปรแกรมบนดอสที่มีอยู่มากไปใช้บน RISC ก็ไม่ใช่เรื่องง่าย นอกจากตัว RISC เองจะไม่คอมแพตทิเบิลกับ CISC (Complex Instruction Set Computer) ที่ใช้อยู่ก่อนแล้ว เครื่องที่ใช้ซีพียู RISC ยังทำงานบนระบบปฏิบัติการยูนิกซ์ซึ่งแตกต่างจากดอสอย่างสิ้นเชิง

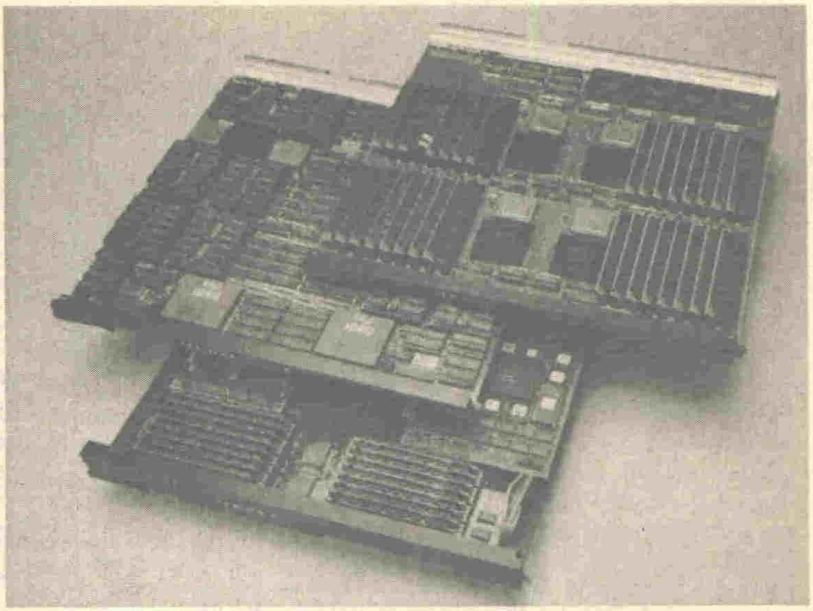
ในขณะเดียวกัน ผู้ใช้พีซีนับล้าน ๆ คนต่างก็ประสบกับปัญหาขีดจำกัดของดอส และตัวเครื่องคอมพิวเตอร์ที่ใช้ดอส เช่น ไอบีเอ็ม พีซี/เอกซ์ที และเอที ซึ่ง RISC เวอร์กสเตชันอาจจะเป็นทางแก้ปัญหานี้ได้ แต่เราก็ประสบกับปัญหาเนื่องจากความไม่สามารถเข้ากันได้ เราจะทำอย่างไรเพื่อแก้ปัญหานี้

การโอนย้ายโปรแกรม

เป็นเวลาอย่างน้อย 25 ปีมาแล้วที่มีการนำวิธีการสองวิธีมาใช้การนำโปรแกรมของเครื่องคอมพิวเตอร์เครื่องหนึ่งไปรันบนเครื่องคอมพิวเตอร์อีกเครื่องหนึ่งที่แตกต่างกัน วิธีแรกคืออีมูเลชัน (Emulation) จะเป็นการจำลองการทำงานของเครื่องคอมพิวเตอร์เครื่องแรกบนเครื่องคอมพิวเตอร์เครื่องที่สอง อีกวิธีหนึ่งก็คือ พอร์ตติ้ง (Porting) จะเป็นการย้ายซอร์สโปรแกรมจากเครื่องคอมพิวเตอร์เครื่องแรกมาทำการแก้ไขแล้วคอมพายล์ใหม่บนเครื่องคอมพิวเตอร์เครื่องที่สอง ปัญหาของทั้งสองวิธีการซึ่งมีมาตลอด 25 ปีก็คือ อีมูเลชันทำงานช้าและพอร์ตติ้งต้องใช้เวลาในการทำมาก

อีมูเลชันจะทำงานคล้ายกับอินเตอร์พรีตเตอร์ โดยจะทำการอ่านคำสั่ง (fetch) ทำการแปลคำสั่ง แล้วทำงาน ซึ่งการกระทำแบบนี้จะต้องใช้ชุดคำสั่งของ RISC จำนวนมากต่อคำสั่งของ 8086 แต่ละคำสั่ง ซึ่งโดยเปรียบเทียบแล้วการจำลองการทำงานของโปรแกรมบนดอสบนเครื่อง RISC จะช้ามาก และเนื่องจากเราจะต้องแปลทีละคำสั่ง จึงทำให้ไม่สามารถใช้ออปติไมซ์คอมพายเลอร์ได้

นอกจากนี้อีมูเลชันยังมีข้อเสียอื่น ๆ อีก กล่าวคือ เนื่องจากดอสเป็นโปรแกรมแบบผู้ใช้คนเดียว โปรแกรมที่ทำงานอยู่จะเป็นผู้ควบคุมทรัพยากรทั้งหมดของระบบ ซึ่งแตกต่างจากระบบยูนิกซ์ที่เป็นแบบระบบหลายผู้ใช้ โดยวิธีการที่ใช้บนดอสจะเป็นการใช้ระบบอย่างสิ้นเปลืองในระบบยูนิกซ์ ตัวอย่างเช่น การรอรับข้อมูลจากคีย์บอร์ด โปรแกรมบนดอสจะทำการวนลูปคอยอ่านสถานะของคีย์บอร์ดอยู่ตลอดเวลาจนกว่าจะมีการกดแป้นพิมพ์ ซึ่งถ้าใช้วิธีนี้จะเป็นการสิ้นเปลืองเวลาของซีพียูอย่างมากในระบบมัลติทาสก์ นอกจากนี้ยังก่อให้เกิดการใช้งานแคชอย่างไม่มีประสิทธิภาพด้วย



พอร์ตติ้งเป็นวิธีการที่จะต้องให้ผู้ที่จำหน่ายซอฟต์แวร์นั้นทำการแก้ไขจากซอร์สโปรแกรมพอร์ตติ้ง จะเป็นวิธีการที่ดีที่สุดสำหรับผู้ใช้ แต่โชคไม่ดีที่แทบจะไม่มีการโอนย้ายโปรแกรมที่คุณต้องการมายังระบบยูนิกซ์ที่คุณกำลังใช้อยู่เลย ทั้งนี้เนื่องมาจากเหตุผลทางด้านเทคนิคและเศรษฐศาสตร์

ในแง่ของประสิทธิภาพ โปรแกรมที่ทำงานบนดอสส่วนใหญ่จะมีส่วนที่เขียนด้วยภาษาแอสเซมบลี ถ้าเราโอนย้ายโปรแกรม ส่วนที่เขียนด้วยภาษาแอสเซมบลีนั้นจะต้องเขียนใหม่ทั้งหมด ซึ่งผู้จำหน่ายซอฟต์แวร์จะต้องทำการดูแล ให้บริการ และอัพเกรดโปรแกรมอย่างน้อยสองเวอร์ชัน หรือมากกว่านั้น ถ้ามีการโอนย้ายไปยังระบบอื่นอีก ซึ่งการเขียนใหม่นั้นใช้เวลานาน บางทีอาจจะเป็นปี ๆ เลยทีเดียว นอกจากนั้นการที่ต้องมาคอยดูแลโปรแกรมหลาย ๆ เวอร์ชันนั้น จะต้องใช้ค่าใช้จ่ายอย่างมาก

และจะทำไปเพื่ออะไร ในเมื่อมีผู้ใช้ระบบดอสอยู่ถึง 25 ล้านคน ในขณะที่ระบบยูนิกซ์มีผู้ใช้อยู่เพียง 4-5 ล้านคนเท่านั้น และในจำนวนระบบยูนิกซ์ที่ใช้อยู่นั้นยังประกอบด้วยหลายระบบที่แตกต่างกัน ซึ่งหนึ่งระบบจะต้องทำการโอนย้ายหนึ่งครั้ง และในจำนวนผู้ใช้ระบบยูนิกซ์ 5 ล้านคนนั้น จะมีประมาณ 30,000 คนเท่านั้นที่ใช้ RISC เวอร์กสเตชัน จากตัวเลขนี้จึงไม่เป็นที่น่าประหลาดใจเลยว่าทำไมผู้ผลิตซอฟต์แวร์เลือกที่จะปรับปรุงซอฟต์แวร์ที่ตัวเองมีอยู่ หรือเขียนโปรแกรมใหม่ ๆ ขึ้นมา มากกว่าที่จะทำการโอนย้ายโปรแกรมมายังคอมพิวเตอร์ที่ใช้ RISC ที่ทำงานบนระบบยูนิกซ์

กลยุทธ์ไบนารีพอร์ตติ้ง (Binary Porting)

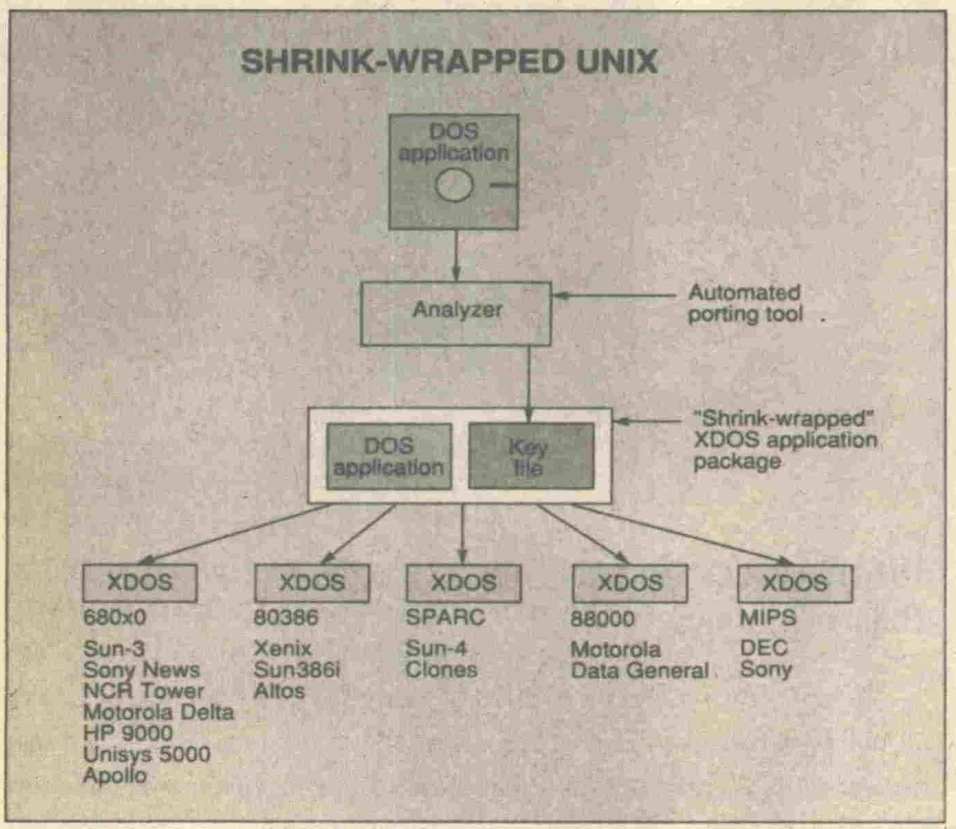
บริษัท ฮันเตอร์ซิสเต็มส์ ได้พัฒนาวิธีการโอนย้ายโปรแกรมแบบใหม่ขึ้นมาเรียกว่า ไบนารีพอร์ตติ้ง ซึ่งไม่ต้องทำการแก้ไขซอร์สโปรแกรมเลย โดยจะใช้เทคนิคของออปติไมซ์คอมพายเลอร์ เข้าไปทำการเปลี่ยนเอ็กซีคิวต์โค้ดของ 8086 ให้เป็นโปรแกรมบนเครื่องที่ต้องการ

ไบนารีพอร์ตติ้งจะสร้างโปรแกรมที่มีประสิทธิภาพและความสามารถในการทำงานเหมือนกับโปรแกรมต้นแบบ และยังใช้เวลาน้อยกว่าในการโอนย้ายซอร์สโปรแกรมโดยตรงด้วย และยิ่งไปกว่านั้น เมื่อทำการโอนย้ายไปยังระบบยูนิกซ์แล้ว โปรแกรมจะสามารถใช้ได้กับเครื่องทุกเครื่องที่ใช้ระบบยูนิกซ์ ทั้ง CISC และ RISC ซึ่งจะทำให้ไม่จำเป็นจะต้องโอนย้ายใหม่เมื่อจะนำไปใช้ยังเครื่องที่ต่างกัน

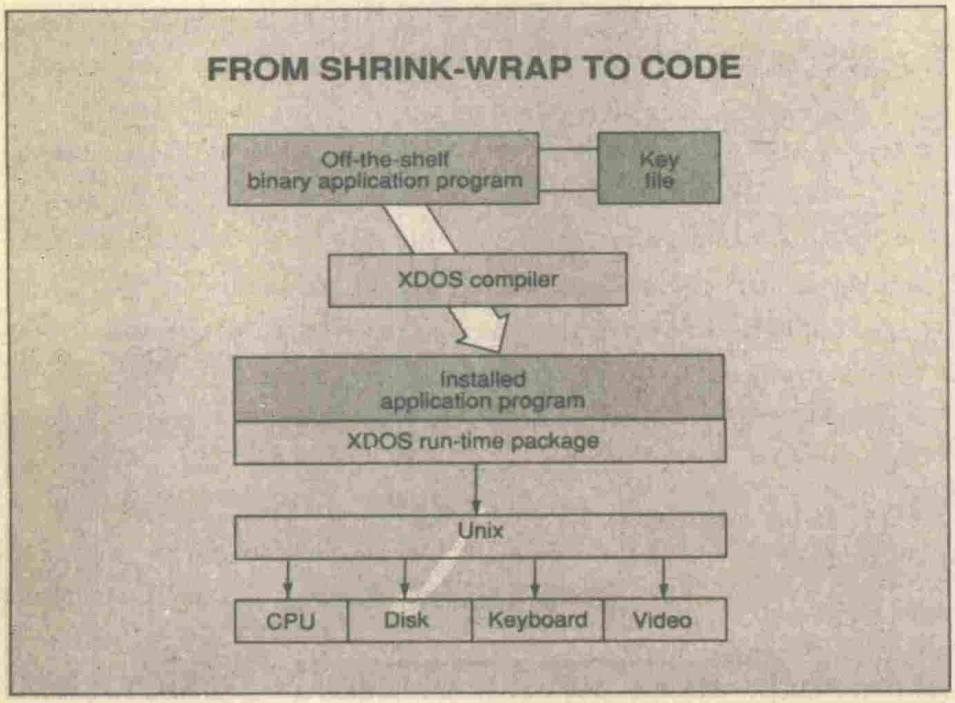
ไบนารีพอร์ตติ้งประกอบด้วย 2 ขั้นตอน ขั้นตอนแรกเรียกว่า ไบนารีอะนาไลซิส (binary analysis) ขั้นตอนนี้จะใช้เวลามาก สามารถเทียบได้กับการโอนย้ายซอร์สโปรแกรม แต่ไบนารีอะนาไลซิสจะใช้เวลาเพียงไม่กี่วันหรือไม่กี่สัปดาห์ แทนที่จะเป็นเดือนหรือเป็นปีเหมือนการโอนย้ายซอร์สโปรแกรม โดยที่ไบนารีอะนาไลเซอร์จะทำการอ่านเอ็กซีคิวต์โค้ดของโปรแกรมที่ทำงานบนดอสมาทำการวิเคราะห์การไหลของข้อมูล (data flow) และจะทำการสร้างไฟล์ที่เรียกว่า คีย์ไฟล์ (key file) เพื่อเก็บผลที่ได้จากการวิเคราะห์

คีย์ไฟล์จะถูกนำไปรวมกับโปรแกรมเดิมเป็นโปรแกรมที่เรียกว่า shrink-wrapped ซึ่งโปรแกรมนี้จะถูกแจกจ่ายไปยังผู้ใช้ระบบยูนิกซ์ ที่ใช้เป็นเครื่องคอมพิวเตอร์แบบใดก็ได้ ไม่ว่าจะเป็น RISC หรือ CISC, System V หรือ Berkeley Unix ซึ่งโปรแกรมนี้จะเป็นยูนิเวอร์แซลไบนารีที่สามารถรันได้บนเครื่องทุกเครื่องที่ใช้ระบบยูนิกซ์ ดังรูปที่ 1

ขั้นตอนที่สองเรียกว่า ไบนารีคอมพลิเลชัน (binary compilation) ซึ่งจะทำครั้งแรกเพียงครั้งเดียวเท่านั้น โดยโปรแกรมที่เรียกว่าไบนารีคอมพายเลอร์จะทำการอ่านโปรแกรมที่



รูปที่ 1



รูปที่ 2

ทำงานบนดอสและคีย์ไฟล์เข้ามา และทำการสร้างเป็นโปรแกรมของ RISC โดยจะใช้ข้อมูลในคีย์ไฟล์เพื่อทำออปติไมซ์โปรแกรมที่ได้ จะเป็นการโปรแกรมที่ทำงานระบบยูนิกซ์จริง ๆ ที่มีประสิทธิภาพเหมือนกับโปรแกรมต้นฉบับ ซึ่งขั้นตอนที่สองนี้อาจจะใส่ไว้ในโปรแกรมติดตั้งระบบก็ได้

รายละเอียดทางโครงสร้างของเครื่องและระบบปฏิบัติการ จะถูกบรรจุไว้ในตัวโปรแกรมไบนารีคอมพายเลอร์ ซึ่งตัวไบนารีคอมพายเลอร์แต่ละตัวจะใช้ได้กับเครื่องในระบบที่มันบรรจุข้อมูลไว้เท่านั้น แต่มันก็ง่ายในการโอนย้ายไปใช้ในเครื่องอื่น ๆ ที่มีโครงสร้างแตกต่างกัน โดยเพียงแต่เปลี่ยนข้อมูลโครงสร้างของเครื่องและระบบปฏิบัติการใหม่เข้าไป ไบนารีคอมพายเลอร์ในแต่ละระบบจะสามารถนำโปรแกรมบนระบบดอสกับคีย์ไฟล์ของโปรแกรมนั้นไปใช้ได้เหมือนกันทุกโปรแกรม

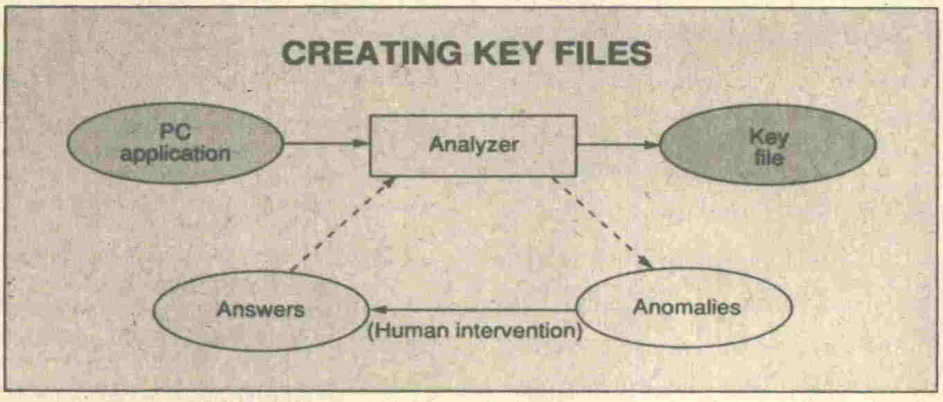
ไบนารีอะนาไลซิส (Binary Analysis)

ปัญหาเบื้องต้นของไบนารีอะนาไลซิส คือการหาเอ็กซีคิวต์โค้ดทั้งหมดของโปรแกรมที่มันวิเคราะห์เพื่อทำการสร้างโฟว์กราฟ (flow graph) ของโปรแกรม ซึ่งจะนำไปใช้เป็นข้อมูลในการสร้างคีย์ไฟล์ต่อไป การหาเส้นทางเพื่อสร้างโฟว์กราฟนี้ ตัวอะนาไลเซอร์จะทำการอ่านคำสั่งทีละคำสั่งโดยเริ่มที่จุดเริ่มต้นของโปรแกรม ซึ่งวิธีการนี้เป็นการง่ายที่จะหาเอ็กซีคิวต์โค้ดทั้งหมด ถ้าไม่มีการเขียนโปรแกรมแบบแก้ไขตัวเองหรือใช้การกระโดด หรือเรียกโปรแกรมย่อยแบบสัมพัทธ์ แต่

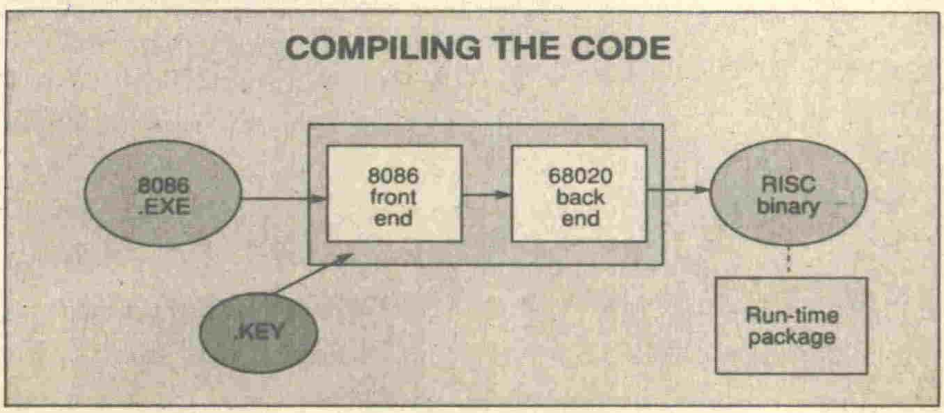
ในความจริงแล้วโปรแกรมส่วนใหญ่จะใช้สองวิธี เพื่อแก้ปัญหานี้ ตัวอะนาไลเซอร์จะทำการสร้างเงื่อนไขสมมติขึ้น ถ้าเงื่อนไขที่สมมติขึ้นเกิดการขัดแย้งกัน อะนาไลเซอร์จะแสดงเงื่อนไขแล้วให้มนุษย์เป็นผู้วิเคราะห์เงื่อนไขนั้นแล้วใส่คำตอบกลับไปยังอะนาไลเซอร์เพื่อช่วยให้การวิเคราะห์ โปรแกรมอะนาไลเซอร์สามารถที่จะทำงานในแบบโต้ตอบกับผู้ใช้ โดยมันจะแสดงถึงคำสั่งของโปรแกรมค่าตัวแปรต่าง ๆ โฟว์กราฟโปรแกรมบางส่วนที่ถูกวิเคราะห์แล้ว และส่วนที่ไม่สามารถวิเคราะห์ได้ ดังรูปที่ 3

เมื่อทำการสร้างโฟว์กราฟเสร็จแล้ว ตัวอะนาไลเซอร์จะทำการวิเคราะห์เพื่อสร้างข้อมูลสำหรับใช้ในขั้นตอนการออปติไมซ์ รวมถึงวิเคราะห์รีจิสเตอร์ที่ใช้และไม่ใช้ และโปรแกรมส่วนที่เป็นเงื่อนไข ซึ่งข้อมูลเหล่านี้จะถูกใช้ในขั้นตอนการออปติไมซ์โดยตัวไบนารีอะนาไลเซอร์และไบนารีคอมพายเลอร์ เพื่อสร้างโปรแกรมที่มีประสิทธิภาพที่สุด

ซึ่งเมื่อขั้นตอนในการิวเคราะห์ทั้งหมดเสร็จเรียบร้อย อะนาไลเซอร์ก็จะสร้างคีย์ไฟล์ขึ้นมา ซึ่งคีย์ไฟล์จะประกอบด้วยส่วนที่จะเป็นสำหรับไบนารีคอมพายเลอร์ นั่นก็คือ ที่ใดบ้างที่ไบนารีคอมพายเลอร์จะต้องทำการแปลโค้ด และข้อมูลนี้จะช่วยให้ไบนารีคอมพายเลอร์ทำงานได้อย่างเชิงเส้น โดยปกติแล้วคีย์ไฟล์จะมีขนาดประมาณ 20 เปอร์เซ็นต์ของโปรแกรมต้นฉบับ



รูปที่ 3



รูปที่ 4

ไบนารีคอมพลิเลชัน (Binary Compilation)

ไบนารีคอมพายเลอร์จะใช้ข้อมูลในคีย์ไฟล์ทำการเปลี่ยนไบนารีโค้ดของโปรแกรมต้นฉบับ ให้เป็นแมชีนโค้ดของเครื่องที่ต้องการ ไบนารีคอมพายเลอร์แต่ละตัวจะถูกออกแบบให้เหมาะสมกับระบบแต่ละระบบ ไบนารีคอมพายเลอร์จะประกอบด้วย 3 ส่วนคือ ส่วนแรก front end ซึ่งเป็นส่วนที่อ่านโปรแกรมบนระบบดอสกับคีย์ไฟล์ ส่วนที่สอง back end ซึ่งจะสร้างและทำการออปติไมซ์โปรแกรม และส่วนสุดท้ายคือ รันไทม์ไลบรารี (runtime library) ซึ่งใช้ในการติดต่อกับระบบยูนิกซ์ดังรูปที่ 4

ขั้นตอนในการทำงานจะเริ่มจากตัวคอมพายเลอร์จะอ่านโปรแกรมต้นฉบับ แล้วทำการเปลี่ยนให้เป็นอินเตอร์มีเดียตโค้ด (intermediate code) โดยใช้ข้อมูลต่าง ๆ ในคีย์ไฟล์ โดยการเปลี่ยนจะทำทีละบล็อก เสร็จแล้วก็จะทำการออปติไมซ์อินเตอร์มีเดียตโค้ดที่ได้ หลังจากออปติไมซ์เสร็จแล้ว คอมพายเลอร์ก็จะทำการสร้างแมชีนโค้ดของเครื่องที่ต้องการจากอินเตอร์มีเดียตโค้ด โดยโปรแกรมโค้ดเยนเนอเรเตอร์จะใช้ข้อมูลในคีย์ไฟล์เพื่อใช้รีจิสเตอร์และแฟล็กต่าง ๆ ให้มีประสิทธิภาพที่สุด ตัดตัวที่ไม่จำเป็นทิ้งไป และพยายามใช้รีจิสเตอร์

ที่สามารถใช้ร่วมกันได้ นอกจากนี้ โค้ดเยนเนเรเตอร์ยังเป็นตัวกำหนดว่าจะใช้รีจิสเตอร์ของเครื่องปลายทางตัวใดแทนรีจิสเตอร์ในโปรแกรมต้นฉบับ

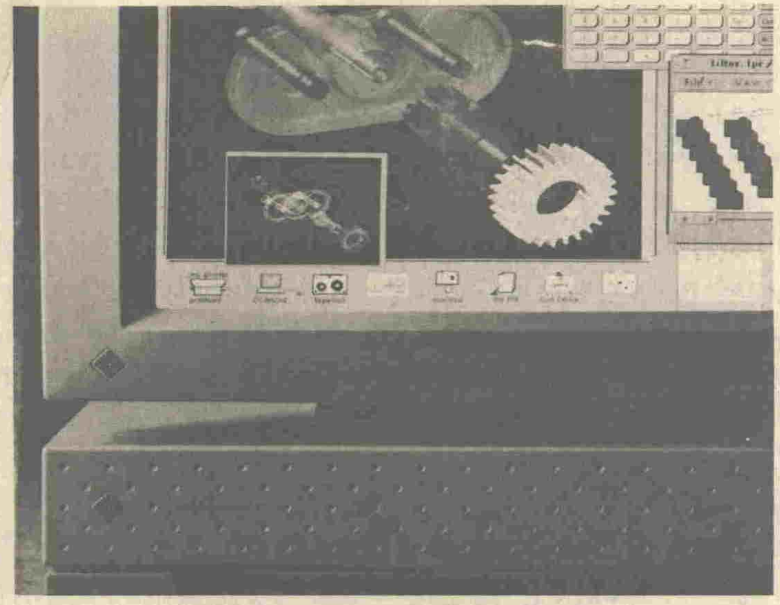
ไบนารีคอมพายเลอร์จะทำการเปลี่ยนส่วนที่เรียกใช้รูทีนของดอสหรือไบออส หรือการเข้าถึงฮาร์ดแวร์โดยตรงเป็นรันไทม์ไลบรารี โดยตัวรันไทม์ไลบรารีนี้จะใช้ทรัพยากรที่มีอยู่ในระบบยูนิกซ์ทำหน้าที่แทน

ออปติไมซ์ RISC โค้ดเยนเนอเรเตอร์

การที่ซีพียู RISC มีประสิทธิภาพเหนือกว่าซีพียู CISC นั้นก็เนื่องมาจากการใช้โค้ดออปติไมเซชัน นั่นก็คือว่า ทำไมอีมูเลชันจึงมีประสิทธิภาพน้อยกว่าโปรแกรมที่ใช้การคอมพายล์ และไบนารีคอมพายเลอร์ก็เป็นคอมพายเลอร์จริง ๆ ที่ใช้เทคนิคของการใช้ออปติไมเซชัน และโค้ดเยอเนอเรชันเหมือนกับคอมพายเลอร์อื่น ๆ และการใช้ไบนารีคอมพายเลอร์ก็ทำได้ง่ายกว่าการโอนย้ายซอร์สโปรแกรมมาทำการคอมพายล์ใหม่

การออปติไมซ์จะเริ่มด้วยการออปติไมซ์แบบง่าย ๆ เช่น การขจัดโปรแกรมในส่วนที่ไม่มีผลต่อการทำงาน หรือรวมคำสั่งย่อย ๆ หลายคำสั่งให้เป็นคำสั่งเดียว ต่อมาคอมพายเลอร์จะทำ dynamic register allocation ซึ่งมีพื้นฐานจากเวอร์ชวลรีจิสเตอร์โมเดล (vitual register model) คอมพายเลอร์จะต้องกำหนดรีจิสเตอร์ของเครื่องที่ย้ายไปขึ้นมาแทนทั้งรีจิสเตอร์ของ 8086 จริง ๆ และรีจิสเตอร์ชั่วคราว (เช่นใช้ในการคำนวณแอดเดรส)

ไบนารีคอมพายเลอร์ยังจะทำ code-motion optimization ซึ่งจะเป็นการขจัดการทำงานซ้ำซ้อน เช่น การใช้คำสั่งกำหนดแอดเดรสซิงโหมดที่ซ้ำซ้อนกัน ซึ่งอาจจะไม่จำเป็นใน 8086 แต่มันเป็นสิ่งที่สำคัญเพราะคำสั่งในการกำหนดแอดเดรสซิงโหมดนั้น จะถูกแทนด้วยคำสั่งของเครื่องอื่น ๆ หลายคำสั่ง



ขั้นสุดท้าย และอาจจะเป็นสิ่งที่สำคัญที่สุดสำหรับเครื่องที่ใช้ซีพียู RISC คือ การจัดสรรตารางเวลาของคำสั่งเพื่อใช้ความสามารถของไปป์ไลน์ดีเลย์ (pipline delay) คอมพายเลอร์จะทำการย้ายคำสั่งคำนวณทางคณิตศาสตร์และคำสั่งทางตรรกมาไว้หลังคำสั่งกระโดด หรือเรียกโปรแกรมย่อยเพื่อที่จะทำให้ดีเลย์สล้อต (delay slot) เต็ม ซึ่งมันจะว่างหลังจากการทำงานคำสั่งกระโดดหรือเรียกโปรแกรมทำงาน

Analyzer และ XDOS

ไบนารีพอร์ตติ้งอาจจะเป็นเพียงแนวความคิดในทฤษฎีเท่านั้น ในทางปฏิบัติแล้วมันจะเป็นอย่างไรเมื่อเทียบกับวิธีการที่ใช้อยู่เดิมคือ อีมูเลชันและพอร์ตติ้ง บริษัท ฮันเตอร์ซิสเต็มส์ ได้ทำการพัฒนาไบนารีพอร์ตติ้งที่สามารถใช้งานได้จริง ๆ ขึ้นมา โดยโปรแกรมอะนาไลเซอร์จะเป็นไบนารีอะนาไลเซอร์ และ XDOS เป็นไบนารีคอมพายเลอร์ และได้ทำการสร้างคีย์ไฟล์ของโปรแกรมบนระบบดอสจำนวนมากขึ้นมา ซึ่งในปัจจุบันสามารถใช้ได้แล้วบนระบบยูนิกซ์

ด้วยโปรแกรมอะนาไลเซอร์ เราสามารถสร้างคีย์ไฟล์ขึ้นมาโดยใช้เวลาน้อยมากเมื่อเทียบกับการโอนย้ายซอร์สโปรแกรมแบบเดิม และคีย์ไฟล์ที่ได้ยังสามารถนำไปใช้กับโปรแกรม XDOS บนเครื่องใดก็ได้

XDOS จะทำการเปลี่ยนโปรแกรมบนระบบดอสให้เป็นโปรแกรมบนระบบที่เราต้องการ บริษัท ฮันเตอร์ซิสเต็มส์ ได้ทำการโอนย้ายโปรแกรม XDOS ไปยังเครื่องที่ใช้ระบบยูนิกซ์หลายแบบ เครื่องที่ใช้ 68000, 80386 MIPS R2000 และ R3000 และ Sun SPARC

รูปที่ 5 แสดงตัวอย่างการย้ายโดย RISC XDOS โดยทำการย้ายจากคำสั่งของ 8086 เป็น 88000 และ R3000 และทำการออปติไมซ์โดยใช้

```
8086 code:

INC.w  ss:0x8e[bp]
MOV    al, ds:0[si]
XOR    ah, ah
PUSH   ax
CALL   L929

Optimized and scheduled intermediate
code:

[0] VR240 :- op_load.h.algn( VRbp
    0x8e )
[1] VRsp  :- op_sub( VRsp 2 )
[2] VR242 :- op_ext16( VRsi )
[3] VR243 :- op_load.b( VRds VR242 )
[4] VR241 :- op_add( VR240 1 )
[5] :- op_store.h.algn( VR241 VRbp
    0x8e )
[6] VR245 :- op_ext8( VR243 )
[7] VRretaddr :- op_bsr( L929 )
[8] :- op_store.h.algn( VR245 VRsp
    0x0 )

Motorola 88000 code:

ld.hu  r2,r18,0x8e
subu   r19,r19,2
extu   r3,r20,16<0>
ld.bu  r3,r25,r3
addu   r2,r2,1
st.h   r2,r18,0x8e
extu   r2,r3,8<0>
bsr.n  L929
st.h   r2,r19,0

MIPS R3000 code:

lhu    $2,0x8e($18)
addiu  $19,$19,-2
andi   $3,$20,0xffff
addu   $1,$25,$3
lbu    $3,0($1)
addiu  $2,$2,1
sh     $2,0x8e($18)
andi   $2,$3,0xff
jal    L929
sh     $2,0($19)
```

รูปที่ 5

instruction scheduling และ dynamic register allocation โปรแกรมต้นฉบับจะใช้ 5 คำสั่ง และใช้ 23 ไซเกิลในการทำงานบน 80386 โปรแกรมที่ถูกย้ายไปบน 88000 จะใช้เพียง 9 ไซเกิล และบน R3000 จะใช้เพียง 10 ไซเกิลในการทำงานซึ่งจะเห็นได้ว่าทำงานได้เร็วขึ้นถึง 2 เท่าในสัญญาณนาฬิกาที่เท่ากัน

การโอนย้าย

คีย์ไฟล์ที่มีอยู่ตอนนี้เป็นโปรแกรมที่ใช้กันอยู่อย่างแพร่หลาย ได้แก่ Lotus 1-2-3, DataEase, dBASE III Plus, Microsofl Word, Multimate, Adventage II, Quattro, R:base, BRIEF, Sprint, WordPerfect 4.2 และ 5.0, WordStar Professional, และ XyWrite III Plus. และบริษัทฮันเตอร์ซิสเต็มส์ ยังจำหน่าย Analyzer ให้กับผู้จำหน่ายซอฟต์แวร์อีกด้วย

นอกจากนี้ บริษัทฮันเตอร์ซิสเต็มส์ ยังได้ส่ง Analyzer และ XDOS ไปยัง Open Software Foundation (OSF) เพื่อใช้เป็น Architecturally Neutral Distribution Format หรือ ANDF เพื่อใช้สำหรับผู้จำหน่ายซอฟต์แวร์สามารถจัดส่งโปรแกรมไปยังผู้ใช้ที่ใช้ระบบยูนิกซ์ที่ต่างกันได้โดยไม่ต้องส่งซอร์สโค้ดไปด้วย (เนื่องจากเครื่องแต่ละเครื่องมีสถาปัตยกรรมและระบบปฏิบัติการที่แตกต่างกัน แต่เดิมโปรแกรมบนระบบยูนิกซ์ ผู้จำหน่ายจะส่งซอร์สโค้ดที่เป็นภาษาระดับสูงไปยังผู้ใช้ และเมื่อผู้ใช้ได้รับซอร์สโค้ดก็จะทำการคอมพายล์เป็นเอ็กซีคิวต์ไฟล์บนเครื่องของตนเอง) ซึ่งการส่งซอร์สโค้ดไปยังผู้ใช้นี้มีข้อเสีย 2 ประการคือ หนึ่ง ผู้ใช้จะต้องทำการคอมพายล์โปรแกรมนั้นใหม่ทั้งหมด สอง โปรแกรมนั้นอาจจะประกอบด้วยข้อมูลที่สำคัญ หรือเทคนิคการเขียนโปรแกรมที่ผู้เขียนต้องการจะเก็บเป็นความลับ ถ้าเราใช้คีย์ไฟล์ของ XDOS กับเอ็กซีคิวต์ไฟล์แบบใดแบบหนึ่งสร้างเป็นโปรแกรม ANDF และทำการส่งโปรแกรมที่อยู่ในรูปแบบของ ANDF ไปยังผู้ใช้ เมื่อผู้ใช้ได้รับโปรแกรมก็เพียงแต่ใช้ XDOS ทำการเปลี่ยนโปรแกรมที่อยู่ในรูปแบบ ANDF นั้นให้เป็นเอ็กซีคิวต์ไฟล์บนเครื่องของผู้ใช้ ด้วยวิธีนี้จะทำให้ผู้จำหน่ายซอฟต์แวร์ไม่จำเป็นจะต้องส่งซอร์สโค้ดไปให้กับผู้ใช้อีกต่อไป

สรุป

ซีพียูขนาด 32 บิต รวมทั้งซีพียู RISC กำลังเริ่มเข้ามามีบทบาทในเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ ดังจะเห็นได้จากเครื่องของบริษัท Digital Equipment (DEC Station 2100), DataGeneral (Aviion), IBM RISC/6000, และ Sun Microsystems (SPARC Station I) ซึ่งแสดงให้เห็นถึงความก้าวหน้าของเครื่องคอมพิวเตอร์บนระบบยูนิกซ์ที่ใช้ RISC ความสำเร็จในการตลาดของเครื่องที่ RISC บนระบบยูนิกซ์รุ่นใหม่ ๆ จำเป็นต้องใช้โปรแกรมที่มีอยู่มากมายบนดอส และทั้งผู้ใช้และผู้จำหน่ายซอฟต์แวร์ก็ต้องใช้โปรแกรมที่มีอยู่แล้ว ไปใช้บนเครื่องที่มีประสิทธิภาพสูงกว่า ซึ่งไบนารีพอร์ตติ้งจะสามารถแก้ข้อบกพร่องต่าง ๆ ดังที่มีในอีมูเลชันและซอร์สพอร์ตติ้งที่ใช้อยู่